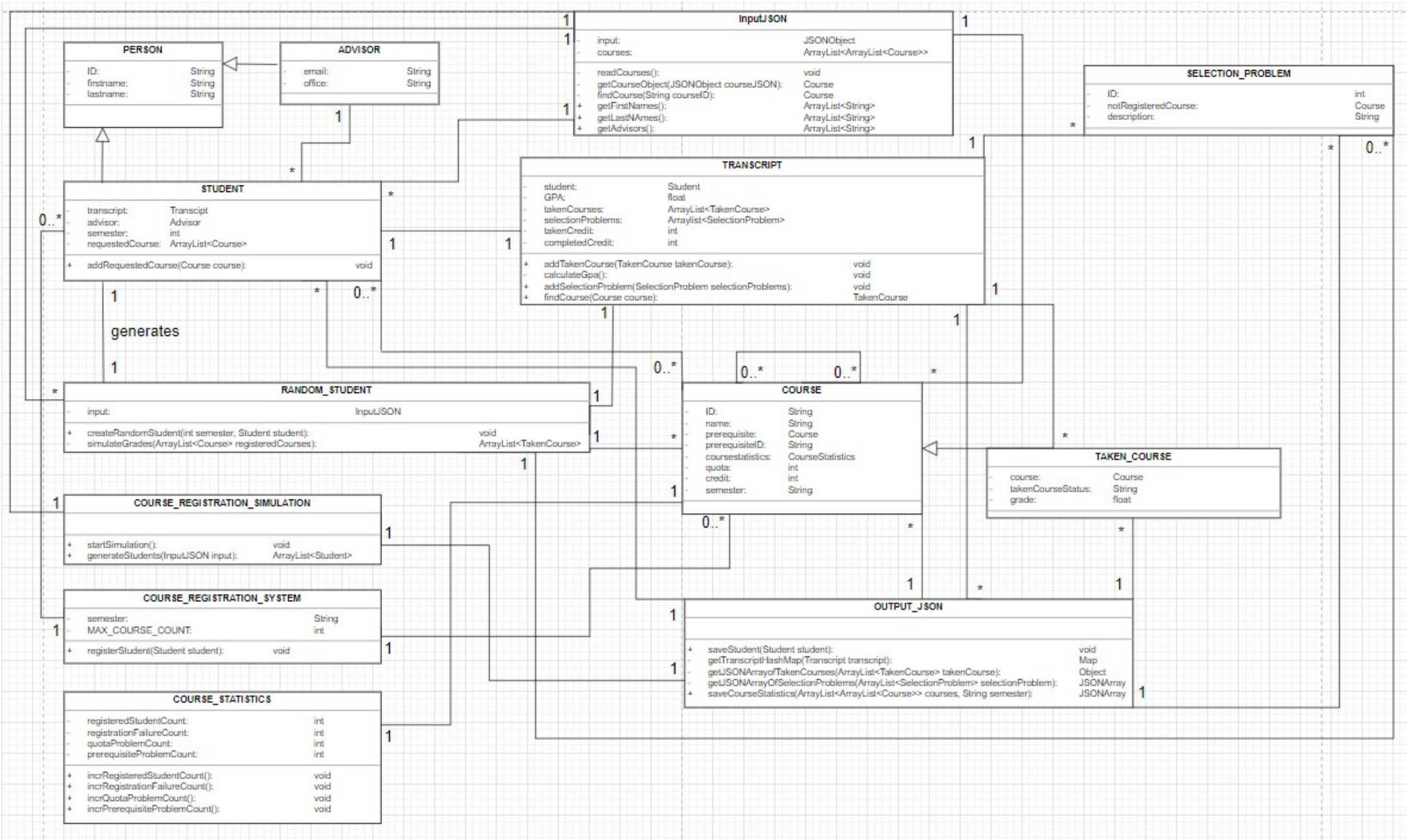
PERSON
- ID: String
- firstname: String
- lastname: String
ADVISOR
- email: String
- office: String
InputJSON
- input: JSONObject
- courses: ArrayList<ArrayList<Course>>
- readCourses(): void
- getCourseObject(JSONObject courseJSON): Course
- findCourse(String courseID): Course
+ getFirstNames(): ArrayList<String>
+ getLastNAmes(): ArrayList<String>
+ getAdvisors(): ArrayList<String>
SELECTION_PROBLEM
- ID: int
- notRegisteredCourse: Course
- description: String
STUDENT
- transcript: Transcipt
- advisor: Advisor
- semester: int
- requestedCourse: ArrayList<Course>
+ addRequestedCourse(Course course): void
TRANSCRIPT
- student: Student
- GPA: float
- takenCourses: ArrayList<TakenCourse>
- selectionProblems: Arraylist<SelectionProblem>
- takenCredit: int
- completedCredit: int
+ addTakenCourse(TakenCourse takenCourse): void
- calculateGpa(): void
+ addSelectionProblem(SelectionProblem selectionProblems): void
+ findCourse(Course course): TakenCourse
generates
RANDOM_STUDENT
- input: InputJSON
+ createRandomStudent(int semester, Student student): void
- simulateGrades(ArrayList<Course> registeredCourses): ArrayList<TakenCourse>
COURSE
- ID: String
- name: String
- prerequisite: Course
- prerequisiteID: String
- coursestatistics: CourseStatistics
- quota: int
- credit: int
- semester: String
TAKEN_COURSE
- course: Course
- takenCourseStatus: String
- grade: float
COURSE_REGISTRATION_SIMULATION
+ startSimulation(): void
+ generateStudents(InputJSON input): ArrayList<Student>
COURSE_REGISTRATION_SYSTEM
- semester: String
- MAX_COURSE_COUNT: int
+ registerStudent(Student student): void
OUTPUT_JSON
+ saveStudent(Student student): void
- getTranscriptHashMap(Transcript transcript): Map
- getJSONArrayofTakenCourses(ArrayList<TakenCourse> takenCourse): Object
- getJSONArrayOfSelectionProblems(ArrayList<SelectionProblem> selectionProblem): JSONArray
+ saveCourseStatistics(ArrayList<ArrayList<Course>> courses, String semester): JSONArray
COURSE_STATISTICS
- registeredStudentCount: int
- registrationFailureCount: int
- quotaProblemCount: int
- prerequisiteProblemCount: int
+ incrRegisteredStudentCount(): void
+ incrRegistrationFailureCount(): void
+ incrQuotaProblemCount(): void
+ incrPrerequisiteProblemCount(): void